



# Tecnológico de Monterrey

Métodos Numéricos

Mtro. Adolfo Centeno T

Differential Equations in Action - Lesson 2

Integrantes:

Maximiliano Macotella Nava, Luz Karen Hernández Hernández, Andres Serrato  
Barrera, Julio Rodríguez Salcedo, César David Rosales Álvares, Jim Kevin Holguín  
Rodríguez

## Lesson 2 - Problem Set 1

LESSON 2

Problem Set 1

[VIEW LESSON →](#)

100% VIEWED

[SHRINK CARD](#)

### 2. Lunar Orbit

```
import math
from udacityplots import *

moon_distance = 384e6 # m
def orbit():
    num_steps = 50
    x = numpy.zeros([num_steps + 1, 2])

    for i in range(num_steps + 1):
        angle = 2. * math.pi * i / num_steps
        x[i, 0] = moon_distance * math.cos(angle)
        x[i, 1] = moon_distance * math.sin(angle)
    return x
x = orbit()
@show_plot
def plot_me():
    matplotlib.pyplot.axis('equal')
    matplotlib.pyplot.plot(x[:, 0], x[:, 1])
    axes = matplotlib.pyplot.gca()
    axes.set_xlabel('Longitudinal position in m')
    axes.set_ylabel('Lateral position in m')
plot_me()
```

Lesson 2:  
Problem Set 1

SEARCH

RESOURCES


CONCEPTS

- ✓ 1. Meet Miriam
- ✓ 2. Lunar Orbit
- ★ 3. Ballistic Trajectories
- ★ 4. Spaceship Trajectories

### Lunar Orbit

SEND FEEDBACK

```
9
10 ~ for i in range(num_steps + 1):
11     angle = 2. * math.pi * i / num_steps
12     x[i, 0] = moon_distance * math.cos(angle)
```



Thanks for completing that!

Correct!

[CONTINUE](#)

[View Intro](#)

[RESET QUIZ](#) [TEST RUN](#) [VIEW ANSWER](#) [SUBMIT ANSWER](#)

[NEXT](#)

### 3. Ballistic Trajectories

```
import math
from udacityplots import *

h = 0.1 # s
g = 9.81 # m / s2
acceleration = numpy.array([0., -g])
initial_speed = 20. # m / s
@show_plot
def trajectory():
    angles = numpy.linspace(20., 70., 6)
    num_steps = 30
    x = numpy.zeros([num_steps + 1, 2])
    v = numpy.zeros([num_steps + 1, 2])
    for angle in angles:
        angle_rad = math.pi / 180. * angle
        x[0, 0] = 0
        x[0, 1] = 0.
        v[0, 0] = initial_speed * math.cos(angle_rad)
        v[0, 1] = initial_speed * math.sin(angle_rad)
        for step in range(num_steps):
            x[step + 1] = x[step] + h * v[step]
            v[step + 1] = v[step] + h * acceleration
        matplotlib.pyplot.plot(x[:, 0], x[:, 1])
    matplotlib.pyplot.axis('equal')
    axes = matplotlib.pyplot.gca()
    axes.set_xlabel('Horizontal position in m')
    axes.set_ylabel('Vertical position in m')
    return x, v
trajectory()
```

The screenshot shows a Udacity course interface. On the left, a sidebar lists 'Lesson 2: Problem Set 1' with a search bar and a list of resources. The main content area is titled 'Ballistic Trajectories' and shows a code editor with Python code. A modal dialog box is centered on the screen, featuring a profile picture of a man, the text 'Thanks for completing that!', and a 'CONTINUE' button. The background is slightly dimmed.

Lesson 2:  
Problem Set 1

Ballistic Trajectories

SEND FEEDBACK

1 # PROBLEM 2  
2 #  
3 # Define the trajectory function below

27 angle\_rad = math.pi / 180. \* angle  
28 x[0, 0] = 0  
29 x[0, 1] = 0.  
30 v[0, 0] = initial\_speed \* math.cos(angle\_rad)

Correct!

## 4. Spaceship Trajectories

```
from udacityplots import *

h = 1.0 # s
earth_mass = 5.97e24 # kg
gravitational_constant = 6.67e-11 # N m2 / kg2
def acceleration(spaceship_position):
    vector_to_earth = - spaceship_position
    return gravitational_constant * earth_mass /
numpy.linalg.norm(vector_to_earth)**3 * vector_to_earth
def ship_trajectory():
    num_steps = 13000
    x = numpy.zeros([num_steps + 1, 2]) # m
    v = numpy.zeros([num_steps + 1, 2]) # m / s
    x[0, 0] = 15e6
    x[0, 1] = 1e6
    v[0, 0] = 2e3
    v[0, 1] = 4e3
    for step in range(num_steps):
        x[step + 1] = x[step] + h * v[step]
        v[step + 1] = v[step] + h * acceleration(x[step])
    return x, v
x, v = ship_trajectory()
@show_plot
def plot_me():
    matplotlib.pyplot.plot(x[:, 0], x[:, 1])
    matplotlib.pyplot.scatter(0, 0)
    matplotlib.pyplot.axis('equal')
    axes = matplotlib.pyplot.gca()
    axes.set_xlabel('Longitudinal position in m')
    axes.set_ylabel('Lateral position in m')
plot_me()
```

The screenshot shows a quiz interface for 'Lesson 2: Problem Set 1' with the title 'Spaceship Trajectories'. A modal window is displayed in the center with a profile picture of a man and the text 'Thanks for completing that!' and 'Correct!'. Below this is a 'CONTINUE' button. The background shows a list of concepts on the left, including '1. Meet Miriam', '2. Lunar Orbit', '3. Ballistic Trajectories', and '4. Spaceship Trajectories'. At the bottom, there are buttons for 'View Intro', 'RESET QUIZ', 'TEST RUN', 'VIEW ANSWER', 'SUBMIT ANSWER', and 'NEXT'.